T.C.
MİLLÎ EĞİTİM BAKANLIĞI

ENDÜSRİYEL OTOMASYON TEKNOLOJİLERİ

GÖMÜLÜ SİSTEMLER 2
523EO0375

Ankara, 2011

- Bu modül, mesleki ve teknik eğitim okul/kurumlarında uygulanan Çerçeve Öğretim Programlarında yer alan yeterlikleri kazandırmaya yönelik olarak öğrencilere rehberlik etmek amacıyla hazırlanmış bireysel öğrenme materyalidir.
- Millî Eğitim Bakanlığınca ücretsiz olarak verilmiştir.
- PARA İLE SATILMAZ.

İÇİNDEKİLER

AÇIKLAMALAR iii
GİRİŞ 1
ÖĞRENME FAALİYETİ–1 3
1. GÖMÜLÜ SİSTEM KARTI 3
1.1. Gömülü Sistem Kartının Tanıtımı 3
1.1.1. Kredi Kartı Büyüklüğündeki L-Card+16M'in Linux ile Kontrolünün Ana Hatları 3
1.1.2. Özellikler 4
1.2. Gömülü Sistem Kartı İle İletişim 6
1.2.1. Bilgisayar ve L-Card+ Arasındaki Bağlantılar 6
1.2.2. L-Card+ İle Açılış ve Kapanış 6
1.3. Gömülü Sistem Kartı Temel Ayarları 8
1.3.1. Karta Giriş Yapmak 8
1.3.2. Kart Dizinlerinin Kontrolü 8
1.3.3. L-Card+ Üzerinde Ağ Ayarları 8
1.4. Gömülü Sistem Kartına Program Gönderimi 9
1.4.1. Karta Program Aktarımı 9
1.4.2. Programın Çalıştırılması 11
UYGULAMA FAALİYETİ 12
ÖLÇME VE DEĞERLENDİRME 13
ÖĞRENME FAALİYETİ–2 15
2. GÖMÜLÜ SİSTEM KARTI KULLANIMI İÇİN GEREKLİ OLAN İŞLETİM SİSTEMİ MODLARI 15
2.1. Kernel Mod ve User Mod 15
2.2. Giriş/Çıkış Portlarının Kontrol Edilmesi 16
2.3. Gömülü Sistem Kartı Kullanımı İçin Gerekli Olan Aygıt Sürücüleri 16
2.3.1. Aygıt Sürücüsü 16
2.3.2. Aygıt Sürücüsü Çeşitleri 16
2.4. Aygıt Dosyası 17
2.5. Majör ve Minör Numaraları 17
2.6. Yüklenebilir Modüller 18
2.7. GPIO Aygıt Sürücüsü (Genel Amaçlı Giriş/Çıkış Ara Birimi) 18
2.7.1. Aygıt Sürücüsü Komutları 19
2.7.2. GPIO Aygıt Sürücüsünün Yüklenmesi 19
2.7.3. Aygıt Oluşturma 20
2.7.4. GPIO Aygıt Sürücüsünün Kontrol Edilmesi 20
2.8. C Programlama Dili ile Kontrol 21
2.8.1. GPIO Fonksiyonları 21
2.8.2. GPIO'nun Fonksiyonlarla Kullanımı 22
UYGULAMA FAALİYETİ 23
ÖLÇME VE DEĞERLENDİRME 27
ÖĞRENME FAALİYETİ–3 29
3.WEB SAYFASI ÜZERİNDEN HTML VE CGI TABANLI DONANIM KONTROLÜ 29
3.1. Temel Kavramlar 29
3.1.1. WEB Sunucusu 29

3.1.2. WEB Sayfası Görüntüleyicisi 30
3.1.3. HTML Dili 30
3.1.4. CGI Dili 30
3.3. CGI Programlama 32
3.3.1. Giriş 32
3.3.2. Program İçeriği 32
3.3.3. Dosya Uzantısı 33
3.3.4. <Form> Etiketi 36
3.4. CGI Programı ile Web Tabanlı Kontrol 38
3.4.1. Ortam Değişkeni (Environment Variable) 38
3.4.2. Led Kontrol Programı 41
UYGULAMA FAALİYETİ 48
MODÜL DEĞERLENDİRME 51
CEVAP ANAHTARLARI 52
KAYNAKÇA 53

AÇIKLAMALAR

KOD	**523EO0375**
ALAN	**Endüstriyel Otomasyon Teknolojileri**
DAL/MESLEK	**Alan Ortak Modülü**
MODÜLÜN ADI	**Gömülü Sistemler 2**
MODÜLÜN TANIMI	Gömülü sistemler hakkında temel bilgilerin verildiği bir öğrenme materyalidir.
SÜRE	40/32
ÖN KOŞUL	
YETERLİK	Gömülü sistem uygulamaları yapmak
MODÜLÜN AMACI	**Genel Amaç** Gömülü sistem kartını teknik özelliklerine ve programlama tekniklerine uygun olarak kullanabileceksiniz. **Amaçlar** **1.** Gömülü sistem kartında programı hatasız gönderip çalıştırabileceksiniz. **2.** Gömülü sistem kartının yerel ağ üzerinden kontrolünü yapabileceksiniz. **3.** Gömülü sistem kartının internet üzerinden kontrolünü yapabileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	**Ortam:** Elektronik laboratuvarı **Donanım:** Bilgisayar, gömülü sistem deney setleri
ÖLÇME VE DEĞERLENDİRME	Modül içinde yer alan her öğrenme faaliyetinden sonra verilen ölçme araçları ile kendinizi değerlendireceksiniz. Öğretmen modül sonunda ölçme aracı (çoktan seçmeli test, doğru-yanlış testi, boşluk doldurma, eşleştirme vb.) kullanarak modül uygulamaları ile kazandığınız bilgi ve becerileri ölçerek sizi değerlendirecektir.

GİRİŞ

Sevgili Öğrenci,

Gömülü sistemler donanımsal ve yazılımsal birçok teknolojinin bir arada kullanılmasıyla oluşturulur.

Araba elektroniği ve ev elektroniği gömülü ürünlere örnek olarak gösterilebilir. Eğer siz de bir gömülü sistem tasarımı yapmak isterseniz bu, geniş çaplı bir bilgi birikimini gerekli kılmaktadır. Elektronik devre bilgisi, mantık devreleri bilgisi, mikro işlemci, donanım mimarisi, programlama ve sistem geliştirme süreçleri hakkında bilgi sahibi olmanız gerekmektedir.

Bu bölümde, gömülü sistem için ihtiyaç duyulan donanımsal ve yazılımsal bilgi ve gereksinimler hakkında bilgi edineceksiniz.

ÖĞRENME FAALİYETİ–1

AMAÇ

Gömülü sistemlerin genel yapısı hakkında bilgi sahibi olacaksınız.

ARAŞTIRMA

- Günümüz bilgisayarlarında kullanılan anakartların hangi özelliklerinin devre dışı bırakılarak bu kartların boyutlarının küçültülmesinin mümkün olup olamayacağını düşününüz. Bunları sınıf ortamında tartışınız.

1. GÖMÜLÜ SİSTEM KARTI

Gömülü sitemler günümüzde birçok endüstri alanında kullanılmaktadır. Bu bölümde seçtiğimiz bir gömülü sistem kartı ile ilgili uygulamalar yapacağız.

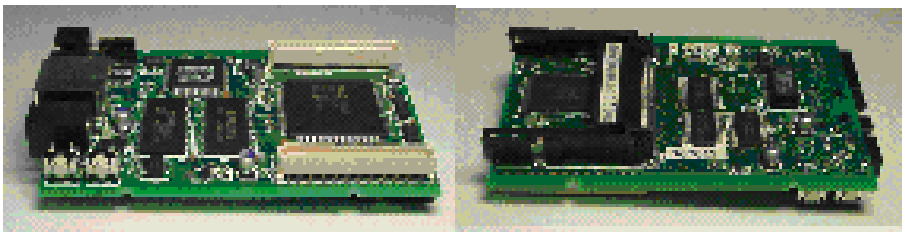
1.1. Gömülü Sistem Kartının Tanıtımı

Linux ile gömülü sistem uygulamaları geliştirebilmek için kullanılan bir araçtır. Tabii ki, "L-Card+16M" i kullanabilmek için bir de bilgisayara ihtiyaç vardır.

1.1.1. Kredi Kartı Büyüklüğündeki L-Card+16M'in Linux ile Kontrolünün Ana Hatları

- L-Card+16M üzerinde; VR4181-66MHz CPU, 2×64Mbit SDRAM, 16MbitFlashRAM, 10base-T LAN, RS232C×1ch, CompactFlash I/F ve genişletilmiş 64P×2 bus yapısı barındırmaktadır.
- Vr418'in üzerine kurulabilecek taşınabilir bilişim aygıtı için gerekli çevre birimleri aşağıda verilmektedir.
  - Hafıza kontrolcüsü (memory controller)
  - Sistem bus kontrolcüsü (system bus controller)
  - Lcd kontrolcüsü (lcd kontroller)
  - Dokunmatik panel ara birimi (touch panel interface)
  - Dma kontrolcüsü (direct memory access controller)
  - Klavye ara birimi
  - Seri ara birim
  - Gerçek zamanlı saat
  - Ses ara birimi
  - A/D dönüştürücü
  - Güç yönetimi

Bu ve buna benzer bilişimsel aygıtlar tasarlamak için "L-Card+" kullanılabilir.



Şekil 1.1: L-card üst ve alt görünüş

1.1.2. Özellikler

- Çevre donanım aygıtlarının çapraz derleyiciler (cross compiler) kullanarak özelleştirilmesi ve bu bilgilerin FlashRom'a linux çekirdeğinin (kernel) derlenmesi yoluyla yazdırılması mümkündür.
- Telnet Server ve FTP kullanıcı tarafında çalıştığı için ağ üzerinden kullanım mümkün olabilir.
- Linux (Kernel ver.2.4) CF disk üzerinden 'boot' edilebilir. Bu yüzden, CF ile kamera kullanımı, kablosuz ağ kullanımı ve CF harddisk kullanımı gibi uygulamaları aynı anda yapabilmek için CF konnektörünün çoğullanması gereklidir.
- FlashROM 16M gelmektedir and L-Card +2 pastM8 times ve 12MB'lık bir alan kullanıcı alanında flash disk şeklinde kartın üzerinde bulunmaktadır.
- Compact ROM disk üzerinde Web Server (ghttpd) bulunmaktadır (12MB'lık kullanıcı alanı içinde).

CPU	Vr4181 66 MHz made by NEC
SD RAM	16 MB
Flash ROM	16 MB(Görüntüleme ve açılış programları için kullanılır.)
Serial port	RS232Cx1ch (10 pin CN)
LAN	10 BASE-T 1ch (CS8900A-CQ3)
Genişletilmiş BUS	Vr4181 ISA sinyali üretebilir.
Genişletilmiş kart I/F	Compact Flash
Güç Tüketimi	0.55 W (max1.5W) ~ 2.25 W CF takıldığında
Kart Ölçüleri	60 mm x 91 mm
OS	Linux (kernel2.4)

Tablo 1.1: L-Card+16M'in ana hatları

Kartın üzerinde 32 adet GPIO (general purpose I/O-genel amaçlı giriş-çıkış) bulunmaktadır. Bunlar kullanıcı tarafından giriş ve çıkış aygıtlarının bağlanmasında L-Card+'ın izin verdiği ölçüler içinde yapılabilir, bacakların kullanımında asıl ölçüt VR4181 CPU'su olmasına karşın L-Card bazı GPIO'ları özel amaçlarla kullanmaktadır.

L-Card'ın kullanımına izin vermiş olduğu GPIO'lar aşağıdaki listede gösterilmektedir.

Sinyal adı	Sinyalin kullanım amacı	L-Card+'da kullanımı	Seçili	Konnektör	Varsayılan değer
GPIO00	SI		√	CN5-20	Hi-Z
GPIO01	S0		√	CN5-19	Hi-Z
GPIO02	SCK	LED6	√	CN5-18	H
GPIO03	PCS0#	CS of CS8900A	X	CN5-17	-
GPIO04		INTRQ0 of CS8900A	X	CN5-16	-
GPIO05	DCD2#		√	CN5-15	L
GPIO06	RTS2#		√	CN5-12	H
GPIO07	DTR2#	Reserved	X	CN5-11	-
GPIO08	DSR2#	LED5	√	CN5-10	H
GPIO09	CTS2#	Input : CD2	X		-
GPIO10	FRM / SYSCLK	Input : CD1	X		-
GPIO11	PCS1#		√	CN5-07	H
GPIO12	FPD4		√	CN5-38	L
GPIO13	FPD5		√	CN5-37	L
GPIO14	FPD6 / CD1#		√	CN5-36	L
GPIO15	FPD7 / CD2#		√	CN5-35	L
GPIO16	IORD#	IORD	X	CN5-31	-
GPIO17	IOWR#	IOWR	X	CN5-32	-
GPIO18	IORDY	IORDY	X	CN5-34	-
GPIO19	IOCS16#		X	CN5-57	-
GPIO20	UBE# / M		X	CN5-33	-
GPIO21	RESET#	Reset	X	CN5-58	-
GPIO22	ROMCS0#		X	CN6-57	-
GPIO23	ROMCS1#		X	CN6-58	-
GPIO24	ROMCS2#		X	CN6-59	-
GPIO25	RxD1	RS232C	X	CN1-3 ※	-
GPIO26	TxD1	RS232C	X	CN1-5 ※	-
GPIO27	RTS1#	RS232C	X	CN1-4 ※	-
GPIO28	CTS1#	RS232C	X	CN1-6 ※	-
GPIO29	DCD1#	LED3	X		H
GPIO30	DTR1#	Reserved	X		-
GPIO31	DSR1#	LED4	X		H

Tablo 1.2: GPIO (general purpose input/output) port kullanımı

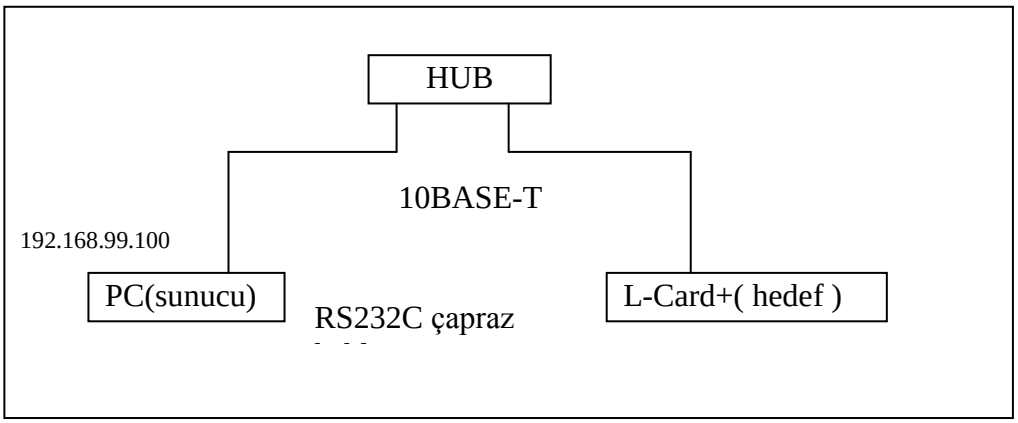
Aktif hâli ‘0’ olan uçlar ‘#’ ile gösterilmiştir.
※ ile gösterilen uçlar seri port bağlantı soketindeki pinlerdir.

1.2. Gömülü Sistem Kartı İle İletişim

Gömülü sistem kartının programlanması için bilgisayarlar iletişimini sağlamak gerekir. Bu bölümde bu iletişim ayarlarını yapacağız.

1.2.1. Bilgisayar ve L-Card+ Arasındaki Bağlantılar

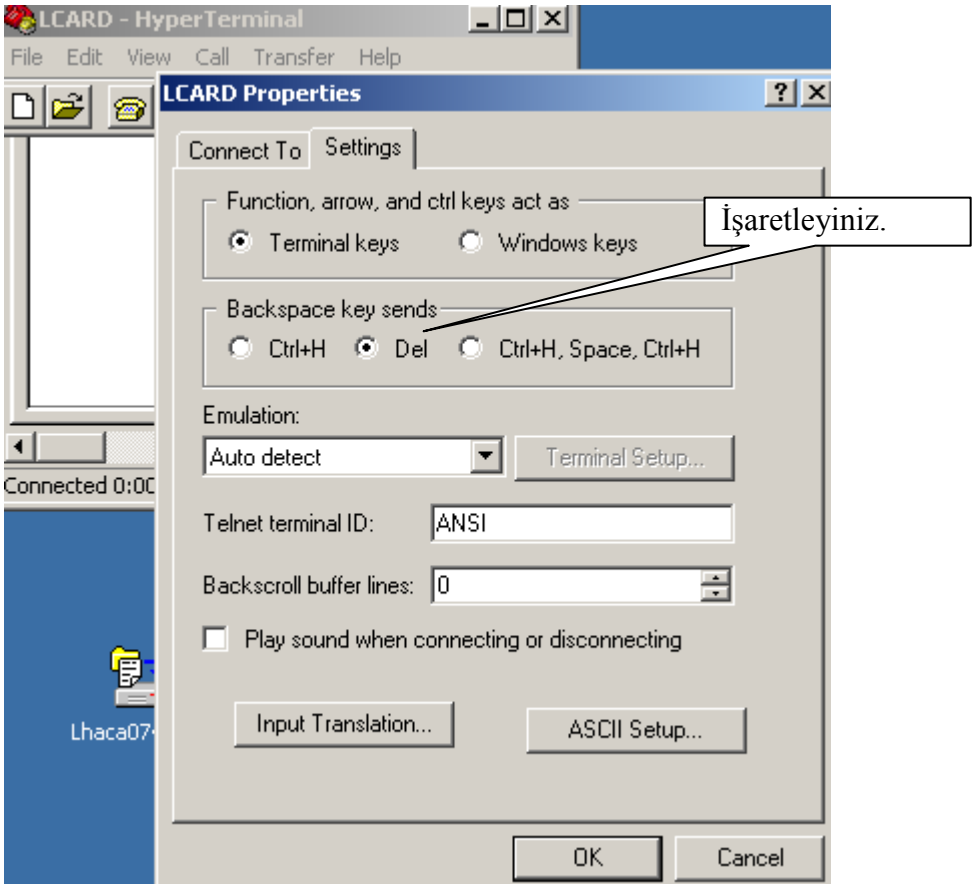
- Bilgisayar ve L-Card birbirine RS232C çapraz kablo kullanarak bağlanır.
- Bilgisayar, L-Card ve Hub bağlantısı 10BASE-T (1:1 kablo) kablo ile yapılır.



Şekil 1.1: L-Card ağ bağlantısı

1.2.2. L-Card+ İle Açılış ve Kapanış

Daha önceden ayarladığımız L-CARD bağlantısı, “Hyper Terminal” programından açılır. “Hyper Terminal” programında aşağıdaki ayarlamalar yapılır.



Şekil 1.2: Terminal bağlantısı

L-CARD'ın beslemesi açılır.
"Hyper Terminal" ekranında aşağıdaki ekran görüntüsü belirir.

waiting...t 0xac0000
==========!
****

[SPACE] tuşuna basarak "monitor mode" seçilir, bu durumda açılış parametrelerini içeren bir menü gelecektir. "linux mode"a geçmek için x tuşuna basılır, bu durumda Linux işletim sisteminin açılış adımları izlenebilir. Eğer bir tuşa basılmazsa "linux mode" otomatik olarak başlatılacaktır.

1.3. Gömülü Sistem Kartı Temel Ayarları

1.3.1. Karta Giriş Yapmak

Giriş yapabilmek için Linux işletim sisteminin açılışını beklemek gerekir. Açılış adımları tamamlandıktan sonra karşımıza gelen ekranda "login:" ifadesi görülür.

login:root (kullanıcı adı:root)
password:laser5(←görünmez)(şifre:laser5)
şeklinde girildikten sonra aşağıdaki ekran karşımıza gelir.

```
(none) login: root
Password:
L-Card+ Release 0.1 2001/12/19
login[25]: root login  on `ttyS0'

L-Card+0.60.1 (2001.12.19-02:47+0000)embeded@laser5.co.jp
#
```

1.3.2. Kart Dizinlerinin Kontrolü

L-card içinde yer alan klasör yapısı standart Linux işletim sistemi klasör yapısıdır. Bunu gözlemlemek için aşağıdaki ekranda verilen işlem adımlarını tek tek uygulamak gerekir.

```
L-Card+0.60.1 (2001.12.19-02:47+0000)embeded@laser5.co.jp
# pwd
/ root
# cd /
# ls –l
bin   etc   lost+found   root   usr ←
boot  home  mnt          sbin   var
dev   lib   proc         tmp ←
```

FLASH ROM alanı

RAM alanı

1.3.3. L-Card+ Üzerinde Ağ Ayarları

L-Card+'a dosya gönderim ve alımını sağlamak için gerekli ayarlamaları yapmak gerekir.

Unutulmaması gereken nokta, ayarlar Ram'da depolandığı için L-Card+ kapatılınca ayarların silinmesidir.

- **Sabit IP ayarları**

ifconfig eth0 192.168.99.101 netmask 255.255.255.0

- **Verilen IP adresinin doğrulanması**

ifconfig

- **Ağ bağlantısını aktif hâle getirmek**

ifconfig eth0 up

- **Ağ bağlantısının denenmesi**

ping - c 3 192.168.99.100

(192.168.99.100 IP'sine üç defa ping yollanır.)
Çalışılan sunucu bilgisayara yollanır.
Altı çizili alan çalışılan host'un IP numarasını gösterir (192.168.99.100)

- **Sonlandırma**

Çıkış # halt
reset # reboot (Ağ bağlantısı sırasında kullanılabilir.)

1.4. Gömülü Sistem Kartına Program Gönderimi

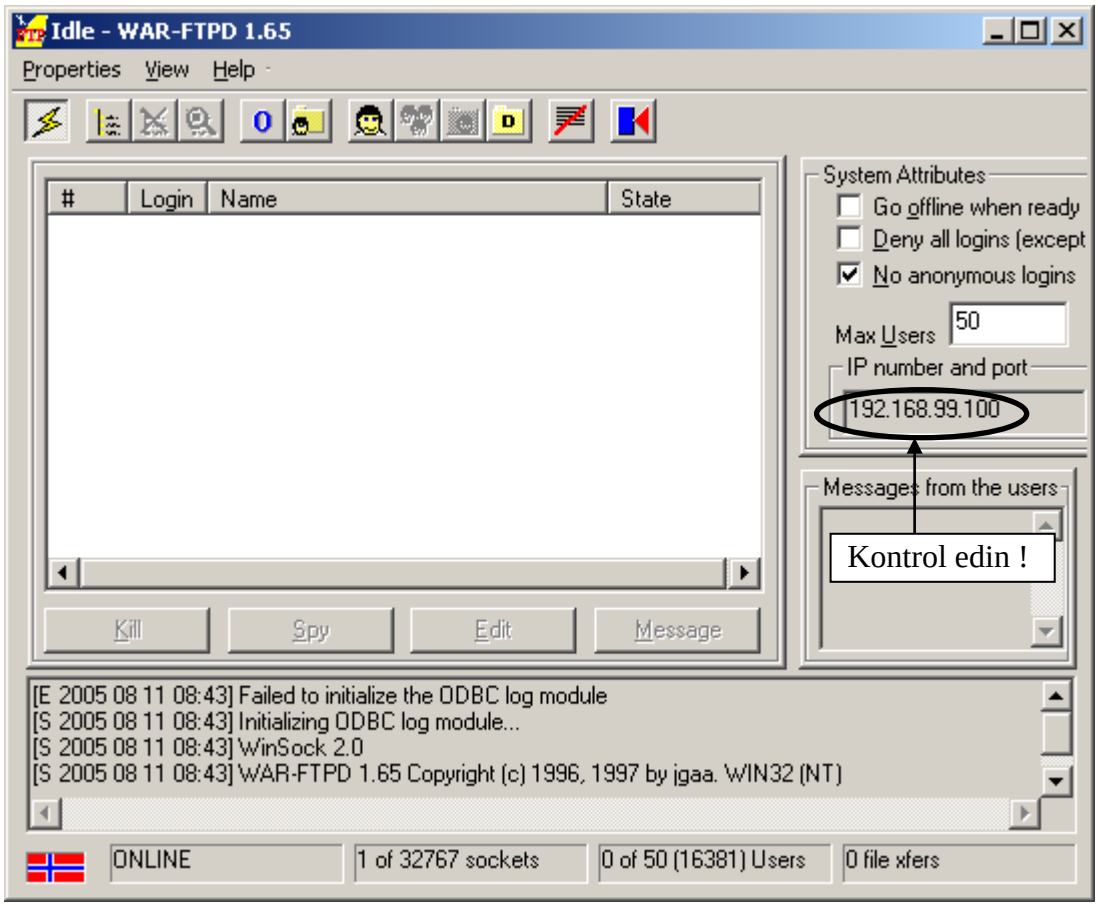
Bu bölüm aşağıdakileri açıklayacak.
- Bilgisayardan L-Card'a dosya aktarılması
- Aktarılan programın çalıştırılması

1.4.1. Karta Program Aktarımı

Karta program aktarımı aşağıdaki adımlar takip edilerek gerçekleştirilir.

- Bilgisayar ve L-Card arasında ağ bağlantısı oluşturulur.

- FTP sunucu başlatılır (War-Ftpd) ve sunucu IP adresi kontrol edilir.



Şekil 1.3: FTP bağlantısı

- Windows'un Command Promptu'ndan FTP sunucuya giriş yapılır.

C\>ftp localhost

User : usr

Password : pass

(görünmez ↑)

- L-Card'da çalışma dizini olarak tmp dizinine geçilir.

```
L-Card+0.60.1 (2001.12.19-02:47+0000)Gömülü@laser5.co.jp
# cd /tmp
```

- Dosya aktarımı gerçekleştirilir.

(Bilgisayardaki FTP sunucudan L-Card'a dosya transferi)

wget komutu:

Bu komut ile FTP sunucudan dosya alınır.

Format:wget ftp:// kullanıcı adı @ Sunucu IP adresi (PC)/ Dosya adı

```
# wget ftp://usr:pass@192.168.99.100/hello-cross
hello-cross  100% |*******************************|   583 KB    00:00 ETA
```

1.4.2. Programın Çalıştırılması

Aşağıdaki işlem adımları gerçekleştirilerek L-Card+'a aktarılan program çalıştırılır.

- Chmod ile çalıştırılabilir dosya iznini ekler (/tmp klasörünün altında olmak şartıyla).

(Chmod komuduna a+x parametreleri eklenir.)

```
# ls –al
drwxr-xr-x    20    0    128 Jan    2 07:08    .
drwxr-xr-x    50    0    160 Jan    1 09:00    ..
-rw-r--r--    10    0    597145 Jan    2 07:08    hello-
cross

# chmod a+x hello-cross
# ls -al
drwxr-xr-x    20    0    128 Jan    2 07:08    .
drwxr-xr-x    50    0    160 Jan    1 09:00    ..
-rwxr-xr-x    10    0    597145 Jan    2 07:08    hello-
cross
```

- Dosya çağırılıp çalıştırılır (/tmp klasörünün altında olmak şartıyla).

```
# ./hello-cross
Hello C World!
```

- FTP sunucu kapatılır. Command Prompt ekranında aşağıdaki komutlar uygulanır.

ftp logout
ftp> quit
(d) Windows'ta "komut prompt"u kapatılır.
C:\> exit

UYGULAMA FAALİYETİ

L-Card'ın bilgisayar ile haberleştirerek derlediğiniz herhangi bir programını L-Card+'a gönderiniz.

İşlem Basamakları	Öneriler
L-Card+'ın bilgisayarınızla haberleşebilmesi için gerekli fiziksel bağlantıları gerçekleştiriniz.	Kullanılacak kablo türüne dikkat ediniz.
L-Card+'ın bilgisayarınızla haberleşebilmesi için gerekli yazılımsal ayarlamaları gerçekleştiriniz (IP adresinin ayarlanması).	L-Card ile bilgisayar IP numaralarının aynı ağda olmasına dikkat ediniz.
Hazırlamış olduğunuz bir programı (çapraz derleyici ile derlenip hazırlanmış) L-Card+'a gönderip gerekli izinleri verdikten sonra çalıştırınız.	Çalışma izinlerini doğru olarak veriniz.

ÖLÇME VE DEĞERLENDİRME

1. Bilgisayarınızın IP adresi 192.168.0.12 ise L-Card+'a aşağıdaki IP adreslerinden hangisini veremezsiniz?

A) 192.168.0.99
B) 192.128.0.12
C) 192.168.0.11
D) 192.168.0.21

2. L-Card'a giriş için kullanacağımız kullanıcı adı ve şifre hangi şıkta doğru olarak verilmiştir?

A) root-laser5
B) main-laser5
C) laser5-root
D) main-root

3. L-Card+'ın IP adresi hangi komutla görülebilir ve değiştirilebilir?

A) ipcfg
B) ipconfig
C) ifcfg
D) ifconfig

4. L-Card+'ı durdurmak için hangi komut kullanılır?
A) stop
B) exit
C) halt
D) reboot

5. Ftp sunucunun başlatılması için verilmesi gereken komut aşağıdakilerden hangisidir?
A) ftp start
B) ftp root
C) ftp localhost
D) ftp run

6. L-Card+'da kullanıcıların girişine izin verilen klasör aşağıdakilerden hangisidir?
A) bin
B) tmp
C) usr
D) localhost

7. FTP'den dosya almak için hangi komut kullanılır?
A) wget
B) get
C) post
D) send

8. Dosyaların kullanıcı izinlerini değiştiren ve dosyaları çalıştırılabilir dosya hâline getiren komut hangisidir?

A) chgmode
B) chmode
C) chmod
D) a+x

9. Dosyayı bulunduğumuz klasörde çalıştırmak için hangi komut verilir?

A) here/dosya adı
B) root/dosya adı
C) /dosya adı
D) /tmp/dosya adı

10. FTP sunucuyu kapatmak için verilmesi gereken komut hangisidir?

A) A)exit
B) B)quit
C) C)cancel
D) D)logout

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

ÖĞRENME FAALİYETİ–2

AMAÇ

Gömülü sistem kartının yerel ağ üzerinden kontrolünü yapabileceksiniz.

ARAŞTIRMA

- Linux işletim sisteminde çalışma modları hakkında bilgi toplayınız.
- Aygıt sürücü ya da "driver" adını verdiğimiz dosyalar hakkında bilgi toplayınız.

2. GÖMÜLÜ SİSTEM KARTI KULLANIMI İÇİN GEREKLİ OLAN İŞLETİM SİSTEMİ MODLARI

Bu bölümde bu modların kullanımı ile ilgili uygulama yapılacaktır.

2.1. Kernel Mod ve User Mod

Unix'te veya Linux işletim sistemlerinde, sistemi kullanıcı programlarından korumak için kullanıcı programlarının giriş/çıkış (I/O) portlarına erişmesine izin verilmez.

İki adet mod bulunmaktadır. Bu modlar Kernel mod ve User mod'dur.

- Kernel mod'unda herhangi bir kısıtlama yoktur. Sistem çağrıları, aygıt sürücüleri ve kernel (işletim sistemi çekirdeği) programı çalışır.
- User mod'unda bazı kısıtlamalar bulunur.

Genel program Kernel (çekirdek) sisteminin çalışmasına izin verir.

Genellikle programlar, Kernel sistemini kullanıcı programlarından korumak için User mod'da çalışır.

Kullanıcı programı bozulsa da Kernel sistemine zarar gelmez.

Kernel mod

Sistem çağrıları, aygıt sürücüsü, kernel program (Tüm bellek adreslerine erişebilir.)

User mod

Genel kullanıcı programları

2.2. Giriş/Çıkış Portlarının Kontrol Edilmesi

Unix ya da Linux işletim sisteminde giriş çıkış portları aşağıdaki gibi kontrol edilir.

- (a)Başka bir kullanıcı tarafından yazılmış I/O aygıt sürücüsü ile
- (b)Yeni bir I/O aygıt sürücüsü oluşturarak

(b) seçeneği çok fazla donanım ve yazılım bilgisi gerektirir ve çok zaman alır.

Ayrıca I/O aygıt sürücüleri Kernel modda çalıştığı için oluşacak herhangi bir sorunda Kernel kendine zarar verebilir. Diğer bir yandan, (a) kendi olağan sürücüsünü kullanır. Daha önceden birçok kullanıcı tarafından denenmiş olduğundan "Kernel"in çalışması zarar vermez.

Bu bölümde açıklanan nedenlerden dolayı (a) seçeneğini uygulayacağız.

2.3. Gömülü Sistem Kartı Kullanımı İçin Gerekli Olan Aygıt Sürücüleri

2.3.1. Aygıt Sürücüsü

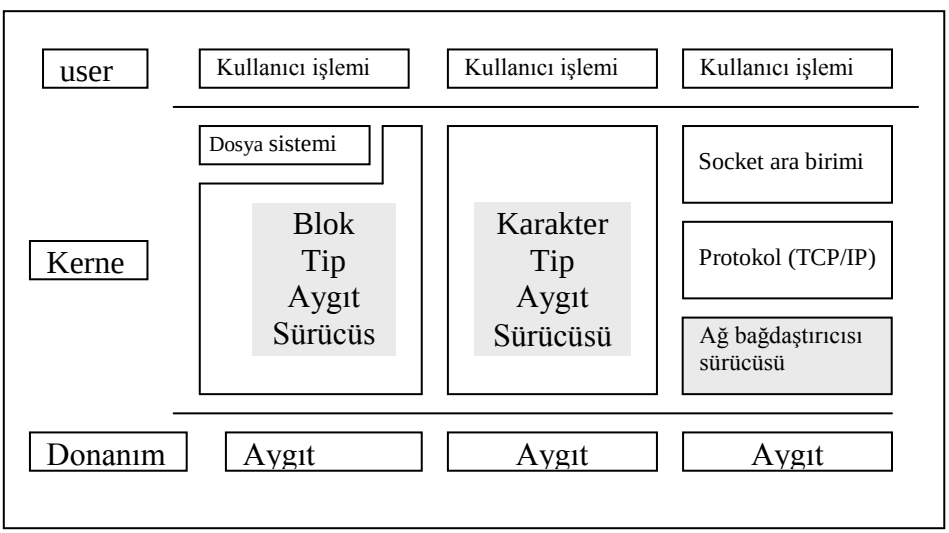
Aygıt sürücüsü programı genellikle sisteme bağlı aygıtları kontrol etmek için kullanılır. Örnek: Klavye, fare, usb, genel amaçlı giriş çıkış birimleri vb. (gpio:general purpose input output)

Aygıt sürücüleri Kernel modda çalışır.

2.3.2. Aygıt Sürücüsü Çeşitleri

Linux'ta üç çeşit aygıt sürücüsü vardır. Bunlar; blok (block) tipi aygıt sürücüsü, karakter (character) tipi aygıt sürücüsü ve ağ bağdaştırıcısı (network interface) tipi aygıt sürücüsüdür.

Bu sürücü çeşitlerinin yapısı aşağıdaki şemada gösterildiği gibidir.



Şekil 2.1: Kullanıcı hakları

2.3.2.1. Blok tip Aygıt Sürücüsü

FD, HD, ram disk, teyp bunun gibi disk sürücüleri blok tip aygıtlardır.
Bir blok boyutu olan 1024, 512 bayt kadar bilgi okunup yazılabilir.
Bu aygıtlar mount edilerek dosya sistemi olarak kullanılabilir.
Blok tip aygıt sürücüsü yapmaya gerek yoktur çünkü tüm blok tip aygıt sürücüleri üreticiler tarafından sağlanmaktadır.

2.3.2.2. Karakter Tip Aygıt Sürücüsü

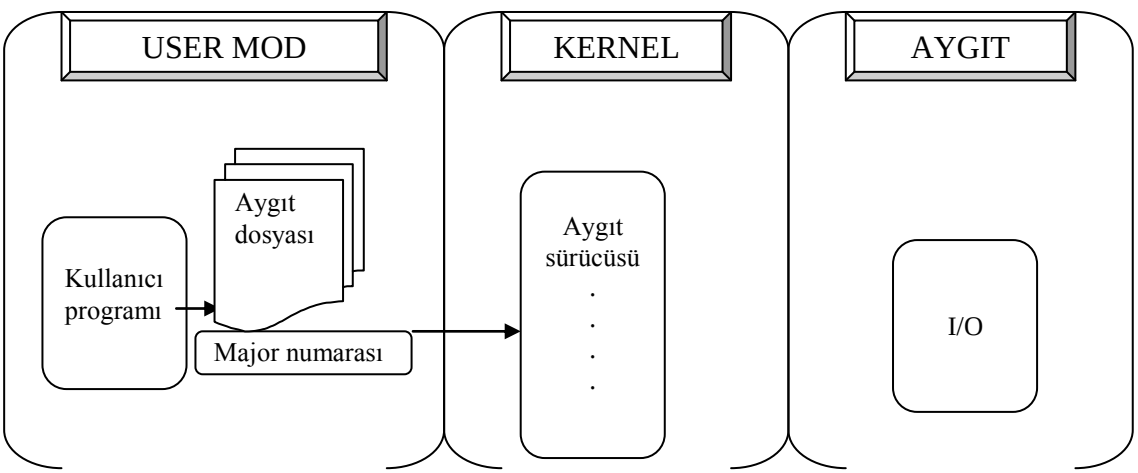
Seri port (tty), paralel port (lp), hafıza aygıtı (mem) karakter tip aygıtlardır.
Bir 1 baytlık bilgi okuyup yazmak mümkündür.
Karakter tip aygıtlar çok esnek aygıtlardır.
Ağ bağdaştırıcısı sürücüleri karakter tip sürücü olarak varsayılır.

2.3.2.3. Ağ Bağdaştırıcısı Tipi Aygıt Sürücüsü

İletişim için ağ bağdaştırıcısı sürücülerini kullanırlar.
Kullanıcı işlemlerinde soket ara birimi olarak kullanılır.
Ağ bağdaştırıcısı sürücüleri protokol katmanı olan TCP/IP (IP paket)'den çağırılır. Ağ bağdaştırıcılarına direkt aygıt dosyasından ulaşılamaz.
Kullanıcı ağ bağlantısını ifconfig eth0…komutu ile başlattıktan sonra soket ara birim programı aygıt sürücüsünü kullanır ver gerekli işlemi gerçekleştirir.

2.4. Aygıt Dosyası

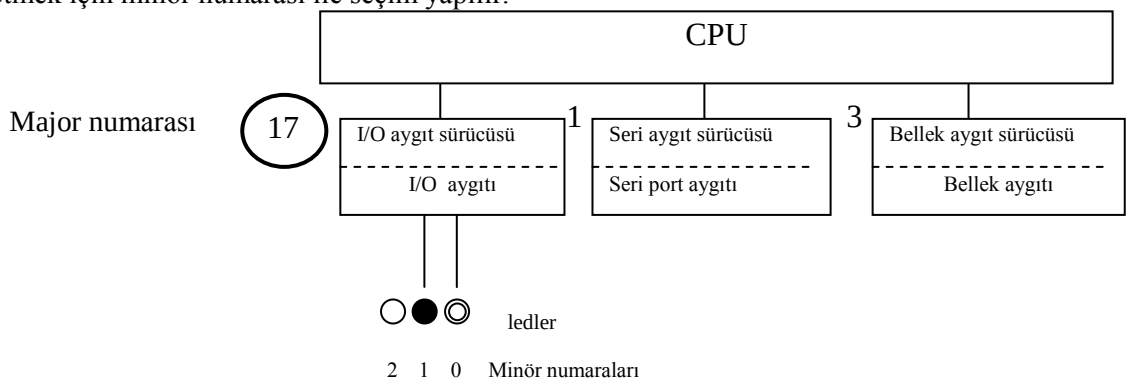
Kullanıcı aygıt sürücüsü kullanmak istendiğinde aygıt dosyası (device file) oluşturulmalıdır.
Bütün aygıt sürücülerinin major (esas) ve minor (ikincil) numaraları vardır.
Şemada da görülebileceği gibi kullanıcı aygıt dosyasına eriştiğinde aygıt dosyasında yazılı olan major numarası aygıt sürücüsünü seçer.



Şekil 2.2: Aygıt dosyası

2.5. Majör ve Minör Numaraları

Aygıt sürücülerinin sıralı özgün majör numaraları vardır. Aynı aygıt içindeki minör numaralarının anlamı birden fazla aygıta ayrılmıştır demektir. Örnek verecek olursak üç adet led var ve bu ledler aynı I/O aygıt sürücüsü tarafından kontrol edilir. İstenilen ledi kontrol etmek için minör numarası ile seçim yapılır.



Şekil 2.3: Majör ve minör numaraları

2.6. Yüklenebilir Modüller

Yüklenebilir modüller Kernel'i geliştiren, Kernel içindeki aygıt sürücüleri gibidir. Diğer bir yandan Kernel'in yeniden derlenmesini gerektirir. Aygıtı her taktığımızda Kernel'in yeniden derlenmesi gerekir, bu hoş bir durum değildir. Aynı şekilde aygıt sürücüsünün versiyonunun yenilenmesi de gerekir.

Tabii ki bunların sonucu olarak yüklenebilir/kaldırılabilir modüllere ihtiyaç duyulmuştur. Yüklenebilir modüller sistemi durdurmadan silinip yüklenebildiklerinden sistem işleyişine uygundur.

2.7. GPIO Aygıt Sürücüsü (Genel Amaçlı Giriş/Çıkış Ara Birimi)

L-Card'da giriş/çıkış aygıtlarından biri GPIO portudur (general purpose ınput/output). L-Card üzerinde 32 GPIO portu bulunur (Tablo 1.2).

Kullanıcı programı GPIO portlarını GPIO aygıt sürücüsü ile kontrol eder.

L-Card üzerinde yer alan LED 3,4,5 ve 6 GPIO portuna bağlıdır. Bu ledleri kullanarak GPIO aygıt sürücüsü hakkında deneyim kazanmak mümkündür.

GPIO portu ve ledler arasındaki ilişki aşağıdaki gibidir.

Minor numarası GPIO port numarası ile aynıdır.

Led	Led3	Led4	Led5	Led6
Port	GPIO29	GPIO31	GPIO8	GPIO2
Minor numarası	29	31	8	2

2.7.1. Aygıt Sürücüsü Komutları

- Önceden oluşturulmuş kernel'e yüklenebilir modül yüklemek

insmod <aygıt sürücüsü dosyasının adı>

- Aygıt dosyası oluşturmak için

mknod <aygıt dosya adı> < tip > [major nu. minor nu.]

Tip

c	karakter aygıt tipi için
b	Blok aygıt tipi için

- Yüklenebilir modülü Kernel'den kaldırmak

rmmod <aygıt sürücüsü dosya adı>

- Dosya sisteminin görüntülenmesi

Df

- Sürücü okuma yazma izinlerinin değiştirilmesi

mount -o remount , ro/ <- sadece okunabilir yapmak (read only)
mount -o remount ,rw/ <- okunabilir / yazılabilir yapmak (read/write)

2.7.2. GPIO Aygıt Sürücüsünün Yüklenmesi

- Gpio.o dosyasını CDROM'dan CYGWIN'in data dizinine kopyalayınız.

```
Administrator @ JICA-3BAC1EFB9E ~/data
$ df
fileSistem   1 k-blocks   used  available  use%  mounted on
c:\ ******    ********   ***   *******   ****   **********
*********     ********   ***   *******   ****   **********
d:              647798   647798   0         0%     /cygdrive/d
$ cp /cygdrive/d/develop/gpio.o        ~/data
$ ls
*****    ****  gpio.o    ****
```

- Hyper terminal üzerinden L-Card'a dosyayı transfer ediniz.

```
#cd /tmp
# wget ftp://usr:pass@192.168.99.100/gpio.o
gpio.o   100%|**************************| 6244 00:00ETA
# ls
gpio.o
```

- gpio.o aygıt sürücüsünü çalıştırılabilir yapıp yükleyiniz.

```
# chmod  a+x  gpio.o
# ls –al
  :
-rwxr-xr-x   1   0   6244  Jan  1  09:00
insmod  gpio.o
```

- Major numarasını kontrol ediniz.

```
# cd  / proc /
# cat devices
Character devices :
1    mem
:
17 gpio
:
```

Major numarası

2.7.3. Aygıt Oluşturma

Aşağıdaki işlem adımlarını izleyerek L-Card üzerinde yer alan Led'lerin kullanımı için gerekli aygıtları oluşturunuz.

```
# cd /tmp
# pwd
/mnt/rm/tmp
# chmod  a+x  gpio.o
# mknod  / tmp / led3  c  17  29
# mknod  / tmp / led4  c  17  31
# mknod  / tmp / led5  c  17   8
# mknod  / tmp / led6  c  17   2
# ls
gpio.o   led3  led4  led5  led6
```

RAM alanı

major

2.7.4. GPIO Aygıt Sürücüsünün Kontrol Edilmesi

Konsoldan kontrol edilmek istendiğinde, her aygıt dosyasına "0" gönderildiğinde led yanar, "1" gönderildiğinde söner.

Ledlerin anot uçları '+' ya bağlı olduğu için "0" gönderildiğinde yanar. Konsoldan karakter göndermek istendiğinde 'echo' komutu kullanılır.

Komutun kullanılışı aşağıdaki gibidir.

Format : ***echo (karakter) > (dosya adı)***

Örnek: Karakter : 0 veya 1 Dosya adı : / tmp / led3

Ledleri kontrol ediniz.

```
# echo 0 > / tmp / led3
# echo 1 > / tmp / led3
# echo 0 > / tmp / led4
:
```

2.8. C Programlama Dili ile Kontrol

Oluşturduğumuz aygıtları "echo" komutunu kullanarak tek tek kontrol etmektense bu aygıtları C programlama dilinin imkânlarını kullanarak kontrol edebiliriz.

Bu bölümde L-Card üzerindeki ledlerin C programları ile kontrolünü inceleyeceğiz.

2.8.1. GPIO Fonksiyonları

Öncelikle GPIO aygıt sürücünün bazı fonksiyonlarını öğrenmemiz gerekiyor.

2.8.1.1. Open ( ) Fonksiyonu

Aygıt dosyasını kullanmak için önce aygıt dosyasının açılması gerekir.

Format: ***open (dosya adı , mod)***

mod : sadece okunabilir O_RDONLY veya 0
sadece yazılabilir O_WRONLY veya 1
okunabilir yazılabilir O_RDWR veya 2

O_ …. , tanımlamalarını kullanmak istiyorsanız dosya kontrol başlığını dâhil etmeniz gerekir (include fcntl.h) (file control header).

Önemli:
open( ) **file handle** değeri ile geri döner.
file handle tam sayıdır ve tanımlanmamıştır.

2.8.1.2. Write( ) Fonksiyonu

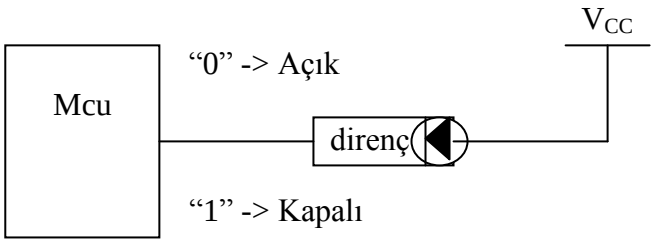
Bu fonksiyon sürücü dosyasını yazmak için kullanılır.

Format: ***write (file_handle, veri , bayt sayısı)***

veri : "0" veya "1"

bayt sayısı : Aygıt dosyasına yazılacak bayt sayısı

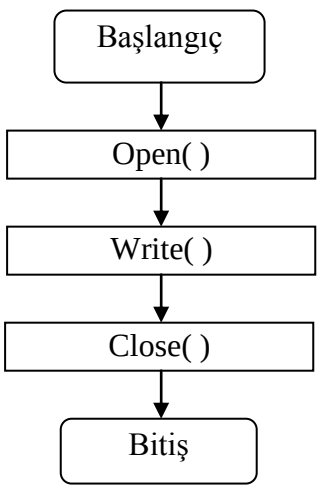
※ Ledleri write( ) fonksiyonu ile aygıt sürücüsüne yazarak kontrol edebilirsiniz.
"0" verisini write fonksiyonu ile gönderirseniz led yanar.
"1" verisini write fonksiyonu ile gönderirseniz led söner.
Çünkü tüm ledler aşağıda gösterildiği gibi bağlıdır.



2.8.1.3. Close() Fonksiyonu

Bu fonksiyon sürücü dosyasını kapamak için kullanılır.

Format:***close (dosya adı)***



2.8.2. GPIO'nun Fonksiyonlarla Kullanımı

GPIO aygıt sürücüsü genel olarak yandaki gibi kullanılır.

UYGULAMA FAALİYETİ

1. Aşağıdaki Led3'ü iki saniyede bir yakıp söndüren programı yazıp çalıştırınız.

İşlem Basamakları	Öneriler
➢ Aşağıdaki programı yazınız.	➢ Bilgisayarda not defterini kullanabilirsiniz.
➢ Programı derleyiniz.	➢ Çapraz derleyiciyi kullanınız.
➢ Derlenen programı L-Card' a gönderiniz.	➢ Wget komutunu kullanınız.
➢ Programın çalışma iznini veriniz.	➢ Chmod komutu ile çalışma izni veriniz.
➢ Programı çalıştırınız.	

```
/*  ledtest1.c */
# include < stdio.h >
# include < fcntl.h >
int  main( )
{
       int  i ,handle[1];
   printf(" LED TEST\n");
   handle[0]=open("led3",O_WRONLY);
   printf("handle[0]=%d\n",handle[0] );

   for ( i =1 ; i<=10 ; i++)
   {
     printf(" times=%d \n", i );

     write( handle[0] , "0" , 1 );
      sleep(1);  /* = = usleep(1000*1000) */

     write( handle[0] , "1" , 1 );
      sleep(1);
    }

      close(handle[0]);

return 0;
}
```

handle[0] Led3'ün aygıt sürücüsü dosyasının file_handle'ıdır.

Sleep (saniye) bu fonksiyon girilen değer kadar gecikme sağlar.

usleep (mikrosaniye) bu fonksiyon girilen değer kadar gecikme sağlar.

Aşağıdaki program Örnek 1'deki programla aynı işlevi gerçekleştirir fakat bu örnek, koşullu operatörlerinin (conditional-operator) kullanılmasını göstermek amacıyla verilmiştir.

Format: ***(verilecek koşul) ? değer1: değer2***

Doğru ise (True) yanlış ise (False)

Eğer koşul doğru ise çıkışa değer 1 aktarılır. Eğer koşul yanlış ise çıkışa değer 2 aktarılır.

```
/*  ledtest2.c */
# include < stdio.h >
# include < fcntl.h >
int  main( )
{
      int  i, handle[1], toggle ;
printf(" LED TEST\n");
handle[0]=open("led3",O_WRONLY);
printf("handle[0]=%d\n",handle[0] );

toggle=1;
for ( i =1 ; i<=20 ; i++)
{

write( handle[0] , toggle  ? "0" : "1" ,1);
sleep(1);

            toggle = !toggle;
}

            close(handle[0]);

      return 0;
}
```

Aşağıdaki program, ledleri değiştirerek yakıp söndürmektedir. Sırasıyla iki programı da deneyebilirsiniz.

```
/*  ledtest3.c */
# include < stdio.h >
# include < fcntl.h >
int main( )
{
       int  d,i ,handle[2] ,toggle ;
    printf(“ LED TEST\n”);
    handle[0]=open(“led3”,O_WRONLY);
    handle[1]=open(“led4”,O_WRONLY);


       toggle=1;
    for ( i =1 ; i<=20 ; i++)
    {
      for(d=0;d<2;d++)
      {
              write( handle[d] , toggle  ? “0” : ”1” , 1 );
        usleep(50*1000);
      }
      toggle = !toggle;
     }
       for(d=0;d<2;d++)  close(handle[d]);
return 0;
}
```

```
/*  ledtest2b.c */
# include < stdio.h >
# include < fcntl.h >
int main( )
{
       int i,t,handle[2] ,toggle;
    printf(“ LED TEST\n”);


handle[0]=open(“led3”,O_WRONLY);
    handle[1]=
open(“led4”,O_WRONLY);

    for ( i =1 ; i<=10 ; i++)
    {
      printf(“ times=%d \n”, i );
     for(t=1;t<=2;t++)
           {
                toggle=1;
      write( handle[0] , toggle ? “0” , 1
);
       sleep(1);  /* usleep(1000*1000) */

      write( handle[0] , “1” , 1 );
       sleep(1);
     }
close(handle[0]);

return 0;
}
```

Bu program ana fonksiyonun argümanını kullanarak aygıt sürücüsünü seçip kontrol yapma olanağı sağlar.

```
/*  ledtest4.c */
#include<stdio.h>
#include<fcntl.h>

int main(int argc ,char *argv[])
{
        int i,d,handle[4],toggle,u,dev_cnt;
        for(u=1;u<argc;ut++)
        {
                handle[u -
1]=open(argv[u],O_WRONLY);
        }
        dev_cnt=argc-1;
           toggle=1;

        for(i=1;i<=50;i++)
        {
                for(d=0;d<dev_cnt;d++)
                {
                        write(handle[d],toggle ? "0" :
"1",1);

                        usleep(50*1000);
                }
        toggle=!toggle;
        }

        for(d=0;d<dev_cnt;dev_cnt++)
        close(handle[d]);

          return 0;
}
```

Programın çalıştırılması

```
#  ./ledtest4  led4  led5  led6
ctrl+c
#  ./ledtest4  led6  led3  led5  led4
ctrl+c
```

ÖLÇME VE DEĞERLENDİRME

1. Aşağıdakilerden hangisi Unix ya da Linux modlarından biridir?

A) User mode
B) Logout mode
C) Linein mode
D) SourceOut mode

2. Aşağıdakilerden hangisi Kernel modda çalışmaz?

A) Sistem çağrıları
B) Aygıt sürücüleri
C) Kernel programı
D) Program kullanıcı arayüzleri

3. Sisteme bağlı aygıtları kontrol etmek için üretici firmalar tarafından oluşturulan ya da kullanıcılar tarafından geliştirilen ve Kernel modda çalışan programlara ne ad verilir?

A) Derleyici programları
B) Algılayıcı programları
C) Aygıt sürücü programları
D) Yükleyici programları

4. Aşağıdakilerden hangisi Linux işletim sistemi altında yer alan aygıt sürücü program türlerinden biri değildir?

A) Kernel tip
B) Blok tip
C) Karakter tip
D) Ağ bağdaştırıcı tipi

5. Kullanıcı programı ile aygıt sürücüsü arasındaki iletişimi kurabilmek için oluşturulan ve kullanıcı programına aygıt sürücüsünün minör ve majör numaralarını bildiren dosyaya ne ad verilir?

A) Aygıt sürücüsü dosyası
B) Aygıt algılama dosyası
C) Aygıt tarama dosyası
D) Aygıt dosyası

6. "Aygıtı kontrol etmek için ……., aygıt birden çok aygıtlara ayrılıyorsa bunları kontrol etmek için bu aygıt altında ……. numaraları kullanılır."
Yukarıdaki cümlede boş bırakılan yerlere aşağıdakilerden hangisi getirilmelidir?

A) Majör, minör
B) Minör, majör
C) Majör, majör
D) Minör, minör

7. Kernel'e yüklenebilir modül yüklemek için kullanılan komut aşağıdakilerden hangisidir?

A) Chmode
B) Insmod
C) Mknod
D) Mount

8. Aygıt dosyası oluşturmak için kullanılan komut aşağıdakilerden hangisidir?

A) Chmode
B) Insmod
C) Mknod
D) Mount

9. Kernel'den yüklenebilir modülü kaldırmak için kullanılan komut aşağıdakilerden hangisidir?

A) Chmode
B) Rmmod
C) Mknod
D) Mount

10. L-Card'a dosya transferi hangi komut ile yapılır?

A) Wpost
B) Wget
C) Get
D) Post

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz

ÖĞRENME FAALİYETİ–3

AMAÇ

Gömülü sistem kartının internet üzerinden kontrolünü yapabileceksiniz.

ARAŞTIRMA

- Web programlama dilleri hakkında bilgi toplayız

3.WEB SAYFASI ÜZERİNDEN HTML VE CGI TABANLI DONANIM KONTROLÜ

Bu bölümde L-Card ile web sayfası üzerinden bir aygıt kontrolü anlatılacaktır.

Ayrıca web sunucuları, web sayfası görüntüleyicileri ve programlaması hakkında bilgi verilecektir. L-Card kullanarak web tabanlı kontrol uygulamaları yapılacaktır.

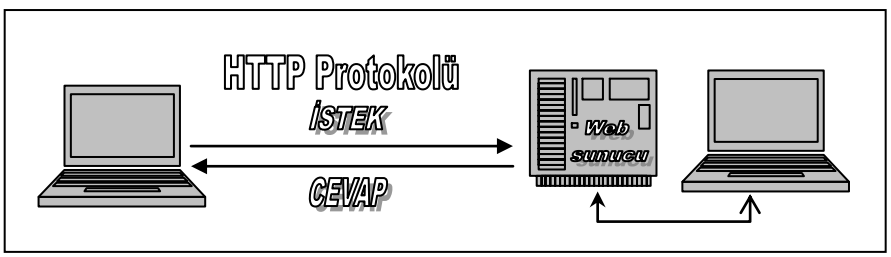
3.1. Temel Kavramlar

3.1.1. WEB Sunucusu

Bütün dünyadaki bilgisayarlar birbirlerine çok çeşitli sunucular ile bağlıdır. Bunlardan web sunucusu, mail sunucusu, FTP sunucusu, dns sunucusu en çok kullanılan sunuculardır.

Herhangi bir web sayfasını web sayfası görüntüleyicimizden yazıp açmaya çalıştığımızda web sunucularına bağlanırız ve web sunucusu bilgisayarımıza web sayfasının dosyalarını gönderir, böylece web sayfalarını görüntüleyebiliriz.

HTTP, web sunucusu ve web sayfası görüntüleyicisi arasında kullanılan protokolün ismidir. Örneğin; http://www.••••••••••••••••••••••, web adresi ya da URL içinde yer alan http ifadesi http protokolünü işaret etmektedir.



Şekil 3.1: Web sunucunun çalışması

3.1.2. WEB Sayfası Görüntüleyicisi

Web sayfası görüntüleyicisi, web sayfalarını görüntüleyebileceğimiz bir programdır. Sık kullanılan programlardan birkaçı Microsoft Internet Explorer, Opera, Firefox, Netscape Navigator'dur.

3.1.3. HTML Dili

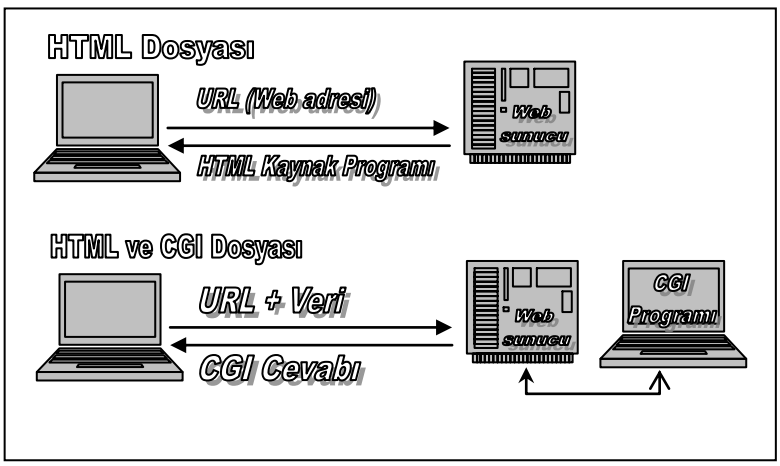
HTML (hyper text markup language) çok basit web sayfası yapabileceğimiz bir dildir. Şimdilerde HTML bilmeyi gerektirmeyen çok fazla web dizayn aracı bulunmaktadır. Fakat HTML kullanmayı öğrenmek web üzerinden kontrol için ve sistemi tanımak için iyi olacaktır.

3.1.4. CGI Dili

CGI (common gateway interface) sunucuda çalışan bir programdır. İnternet dünyasında GCI her ne kadar yerini ASP ve PHP'YE bırakmış olsa da birçok yerde kullanılır. Özellikle basit web tabanlı kontrol sistemlerinde tercih edilir.

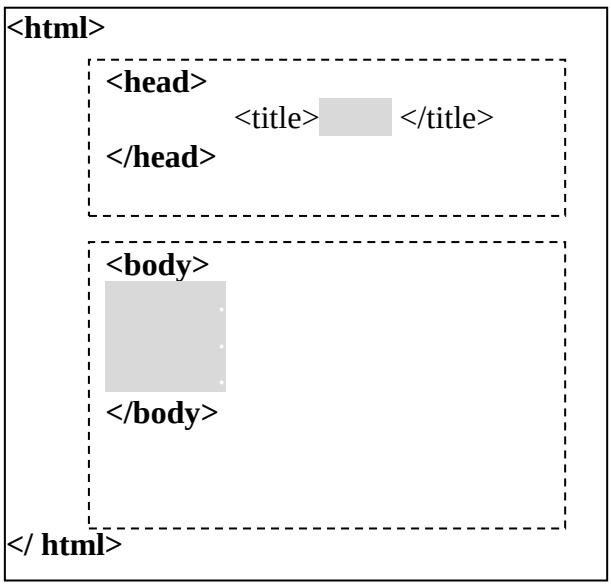
HTML sayfaları sabittir. Yazılar sayfa açıldıktan sonra değişmez.

Ama CGI programları eş zamanlı çalışır yani dinamik bilgi alabilmek mümkündür. Örneğin; CGI programları forumlarda, web sayfası sayıcılarında, satış yapan web sayfalarında kullanılır. Aşağıdaki figürde bir CGI programının işleyişi gösterilmiştir.



Şekil 3.2: HTML ve CGI çalışma prensipleri

3.2. HTML Etiketi (Tag)



Bu bölümde HTML hakkında temel bilgi verilecektir.
HTML dilinde etiket (tag) denilen < >işaretleri kullanılır.
Genel olarak etiketler koyu yazılır <HTML>, <head>, <body>.
Gri ile gösterilen yerlere web sayfası içinde görünmesi istenilen yazılar eklenir.
Ayrıca etiketler yazıların karakterini büyüklüklerini, renklerini, ayrıca web sayfaları ile ilgili artalan rengi gibi birtakım özellikleri değiştirmek için de kullanılır.

Etiket ismi	Özelliği	Etiket ismi	Özelliği
•••	Kalın yazı	 	Bir satır aşağı
•••	Yazı rengi ve büyüklüğü	<A href=filename>•••</A>	Link oluşturmak
<h *>•••</h>	Başlık	•••	Paragraf

*1~6 numaraları arasındaki bir değer olabilir.

Örnek

Not defterini kullanarak aşağıdaki programı yazınız ve ismini HTML.htm olarak kaydediniz.

(dosya ismi : Ders_HTML.html)

```
<html>
  <head>
    <title> HTML Document</title>
  </head>

  <body>
    I am studying HTML now.
    <br>
    <b> HTML </b>
    <font size="+3">is easy</font>
    <p>
    <b> HTML </b>
    <font size="+3">is easy</font>
    </p>
  </body>
</html>
```

Resim 3.1: Browser ekranı

3.3. CGI Programlama

3.3.1. Giriş

CGI ve Perl ortak programlama dilini kullanır. Ayrıca C dilini kullanmak da mümkündür.

CGI programında HTML sayfası oluşturacağınız kısımlar için aşağıdaki ifadeyi kullanmanız gerekmektedir.

printf(“Content-type: text / html\r\n”)

Bu içerik başlığı web browser’a sayfada ne bulunduğunu anlattığı için çok önemlidir.

CGI dili ile kullanabileceğiniz bazı içerik başlıkları şöyledir:

İçerik tipi (Content -type)	**Veri tipi**
Text / plain	Yazı dosyası ise
Resim / jpeg	Jpeg dosyası ise
Resim / gif	Gif dosyası ise

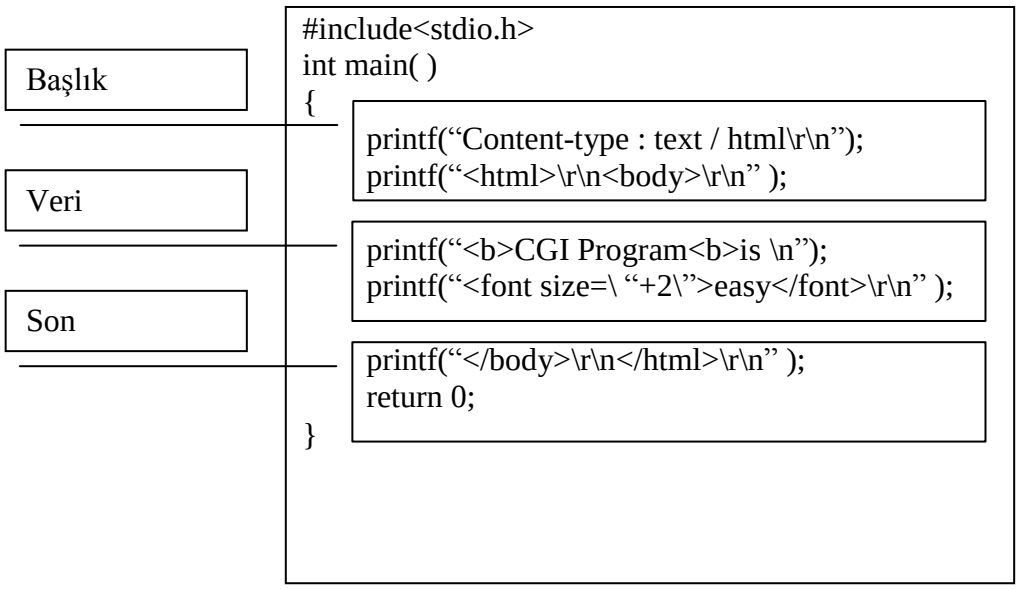
3.3.2. Program İçeriği

Derlenmiş C programı browser’a bu şekilde yardımcı olur.

Yazılmış olan CGI programının browser tarafından anlaşılabilmesi için içeriğin HTML tipinde (HTML type) olması gereklidir.

Programın C dilinde yazılıp derlenmesiyle de CGI dosyası oluşturulabilir. Ancak dosyada birtakım değişiklikler yapılması gereklidir.

C dili ile yazılacak bir CGI programı aşağıdaki gibi üç bölümden oluşur.



- **Başlık**
  Browser'a içerik tipi ve HTML kodunun başlangıcı ile ilgili bilgilendirmede bulunur.
- **Veri**
  Browser'a gönderilecek bilgiler buraya yazılır.
- **Son**
  Browser bu son kısım ile sayfanın bittiğini anlar.

3.3.3. Dosya Uzantısı

C dilinde yazdığımız programlara genellikle (.c) uzantısı verilir. Fakat HTTP sunucu CGI programına ulaşmaya çalıştığında sunucu (.cgi) uzantılı dosyaları arar.

Buna göre yazılan program derlendikten sonra uzantısını –o seçeneğini kullanılarak (gcc derleyicisinde) (.c) ' den (.cgi)' a dönüştürmemiz gerekir.

Örnek 1

Yazı içeren CGI programı (Derleyip L-Card'da çalıştırınız.)

- Aşağıdaki CGI programını VI editöründe yazıp derleyiniz.
- Daha sonra ~/data dizinine kaydediniz (dosya adı : text.c).

```
#include<stdio.h>
int main( )
{
        printf("Content-type: text / plain\r\n");
        printf("\r\n" );
        printf("This is text type cgi program\r\n");
        printf("It is very easy !!!\r\n" );
        return 0;
}
```

- Cygwin' de Mipsel-Linux-Gcc ile derleyip dosyanın uzantısını text.c' den text.cgi' a dönüştürünüz.

```
Administrator @ JICA-3BAC1EFB9E ~/data
$ Mipsel-Linux-Gcc  -O  -static –o  text.cgi  text.c  -s
```

- Dosyaları transfer ediyoruz.

Bilgisayardaki FTP sunucudan L-Card' a wget komutu ile dosyaların Transferi

```
# cd /tmp
# wget ftp://usr:pass@192.168.99.100 / text.cgi
text.cgi  100% |*****************************|    148 KB    00:00 ETA
```

- Çalıştırılabilir izin ekliyoruz.

```
#chmod a+x text.cgi
```

- Programın çalıştırıyoruz.

```
#./text.cgi
This is text type cgi program
It is very easy !!!
#
```

Örnek 2

HTML tipi cgi programı (Derleyip L-Card' da çalıştırınız.)

- Aşağıdaki CGI programını VI editöründe yazıp derleyiniz.
- Daha sonra ~/data dizinine kaydediniz (dosya adı: html. c).

```
#include<stdio.h>
int main( )
{
        printf("Content-type : text / html\r\n");
        printf("<html>\r\n<body>\r\n" );

        printf("<b>CGI Program<b>is \n");
        printf("<font size=\ "+2\">easy</font>\r\n" );

        printf("</body>\r\n</html>\r\n" );
        return 0;
}
```

- Cygwin' de Mipsel-Linux-Gcc ile derleyip dosyanın uzantısını HTML. c' den HTML. cgi'ye dönüştürünüz.

```
Administrator @ JICA-3BAC1EFB9E ~/data
$ Mipsel-Linux-Gcc  -O  -static –o  html.cgi  html.c  -s
```

- Dosyaları transfer ediyoruz.

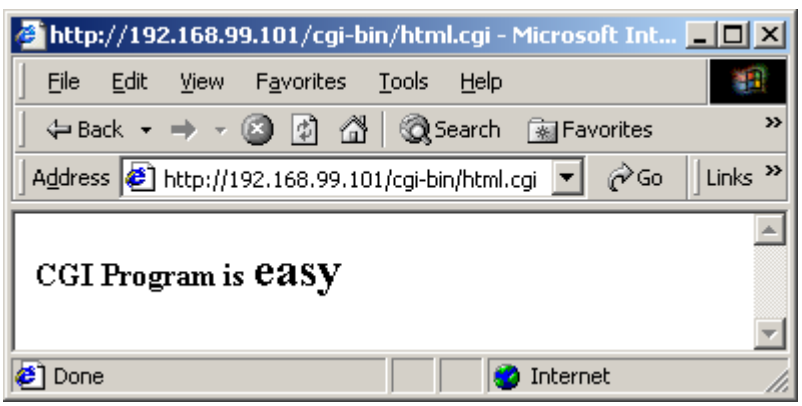
```
# cd /tmp
# wget ftp://usr:pass@192.168.99.100 / html.cgi
html.cgi  100% |*******************************|   148 KB    00:00 ETA
```

Bilgisayardaki FTP sunucudan L-Card' a wget komutu ile dosyaların transferi.

- Çalıştırılabilir izin ekliyoruz.

```
#chmod a+x html.cgi
```

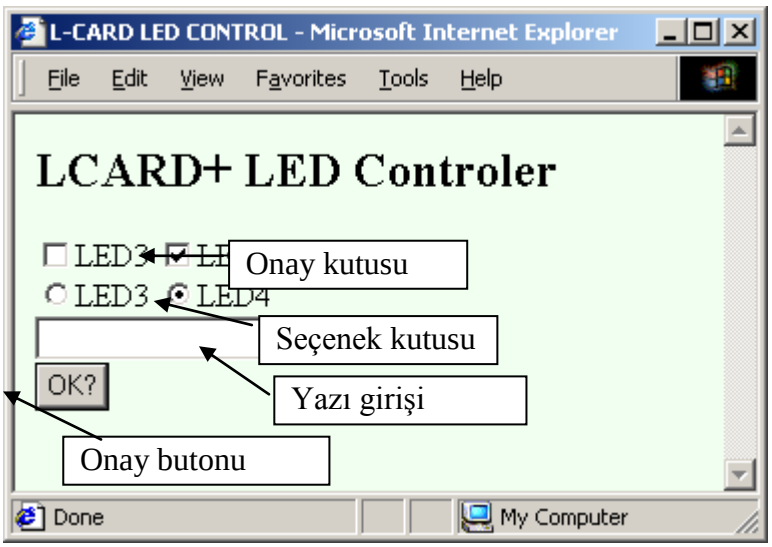
- Program çalıştırıldığında browser'i oradan da http://192.168.99.101/cgi-bin/html.cgi sayfasını açtığımızda sayfa görüntülenir.



Resim 3.2: Programın test edilmesi

3.3.4. <Form> Etiketi

CGI programlarında kullanıcı veri girişi için <form> etiketi sıkça kullanılır.
Bir sonraki resimde de görüldüğü gibi<form> tag'ı ile birçok veri giriş tipi vardır.



Resim 3.3: GET ve POST methodları

<form> tag'ının iki metodu vardır. Bunlar GET ve POST'tur.

3.3.4.1. GET Metodu

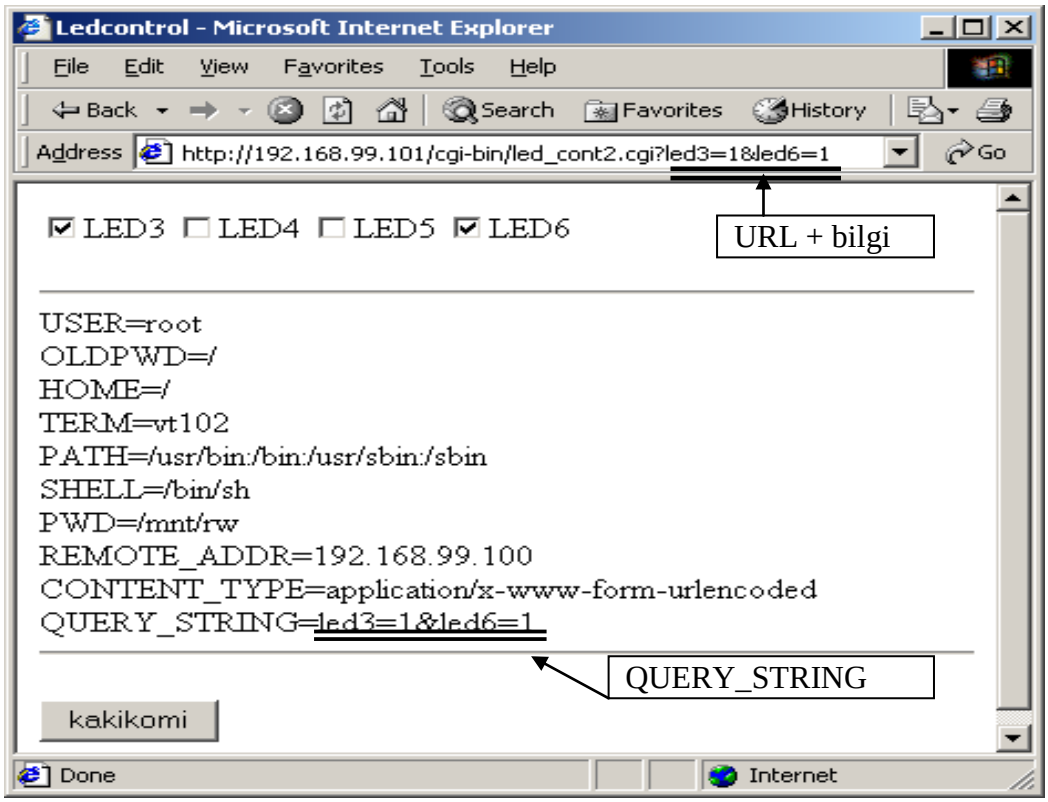
Form tag'ı METHOD="GET" şeklinde tanımlanmış ise CGI programı ile formdan girilen bilgileri, ortam değişkeni QUERY_STRING'ten okunabilir.

3.3.4.2. POST Metodu

Form tag'ı METHOD="POST" şeklinde tanımlanmış ise CGI programı ile formdan girilen bilgileri, ortam değişkeni uzunluğu CONTENT_LENGTH' ten okunabilir.

Örnek 1

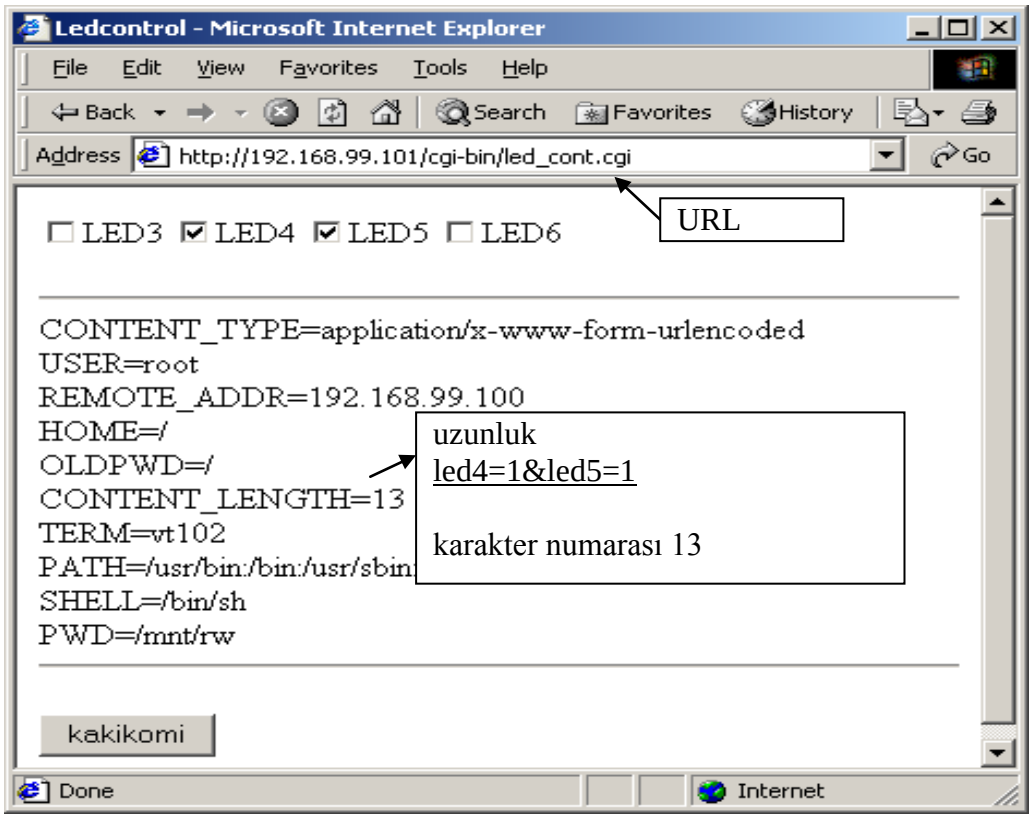
Ortam değişkeni ile GET metodunun kullanımıyla ilgili bir örnek verelim.
URL(uniform resource locator) web sayfası kaynağı



Resim 3.4: GET kullanımına örnek

Örnek 2

Ortam değişkeni ile POST metodunun kullanımıyla ilgili bir örnek verelim.



Resim 3.5: POST kullanımına örnek

POST metodu GET metoduna göre daha güvenlidir. Çünkü POST metodunda sadece URL görülür. Ama POST metoduyla program yapmak çok zordur. Çünkü POST verileri bellekte saklanır. Ana bellek yönetimini iyi bilmek gerekir. Eğer L-Card dış ağa bağlanmayacak ise güvenlik çok önemli olmayacağından POST metodu yerine GET metodu ile yapılabilir.

3.4. CGI Programı ile Web Tabanlı Kontrol

Bu bölümde GET metodunun nasıl kullanıldığını açıklayacağız.

3.4.1. Ortam Değişkeni (Environment Variable)

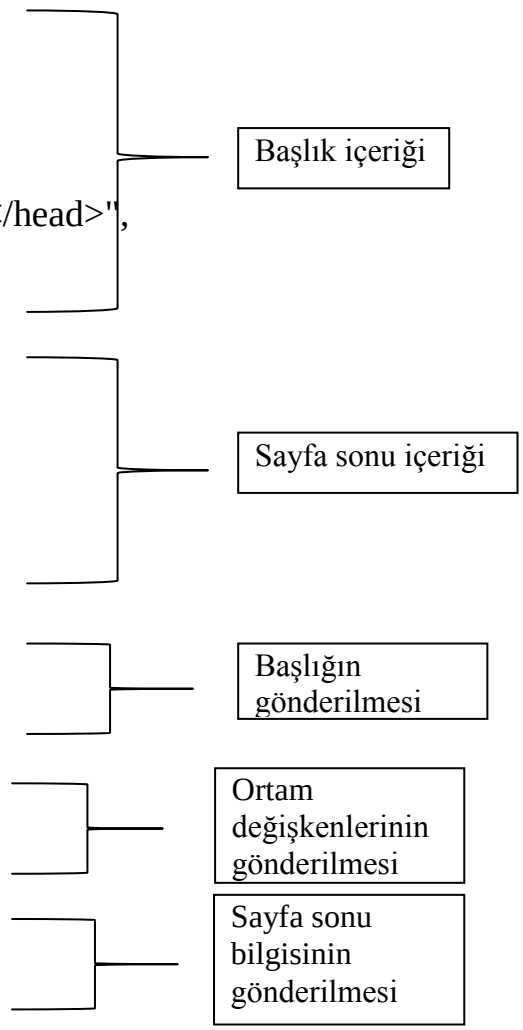
Önce ortam değişkenlerini tanıyabileceğimiz bir örnek program yapacağız.

Ortam değişkenleri dış değişkenlerden (örnekte 'environ') gönderilir. Değişken "pointer"den "yazı verisi dizisi"ne aktarılır. Her yazı değişkeni "CHARACTER STRINGS = veri" şeklinde gelir.

Örnek: CONTENT_LENGTH = 13

Örnek programı yazıp printenv.c dosya adı ile kaydediniz.

```
#include<stdio.h>
int main()
{
        extern char **environ;
        char **pp;
        char *header[]={
          "Content-type: text/html",
        "",
        "<html>",
        "<head><title>printenv</title></head>",
        "<body bgcolor=\"#fff0f0\">",
        NULL};

        char *trailer[]={
        "</body>",
        "</html>",
        NULL,
        };

for(pp=header;*pp;pp++)
        printf("%s\r\n",*pp);

for(pp=environ; *pp;pp++)
        printf("%s<br>\r\n",*pp);

for(pp=trailer;*pp;pp++)
        printf("%s\r\n",*pp);
return 0;
}
```



Açıklamalar

for(pp=header;*pp!=NULL;p++) kodu ile (pp=header;*pp;pp++) kodu birbirine eşittir.

Programı derliyoruz.

```
Administrator @ JICA-3BAC1EFB9E ~/data
$ Mipsel-Linux-Gcc  -O  -static –o  printenv.cgi  print env.c  -s
```

Programı L-Card' a aktarıyoruz.

```
# cd /tmp
# wget ftp://usr:pass@192.168.99.100 / text.cgi
      printenv.cgi  100% |*****************************|  149 KB 00:00 ETA
```

Çalıştırılabilir izin ekliyoruz.

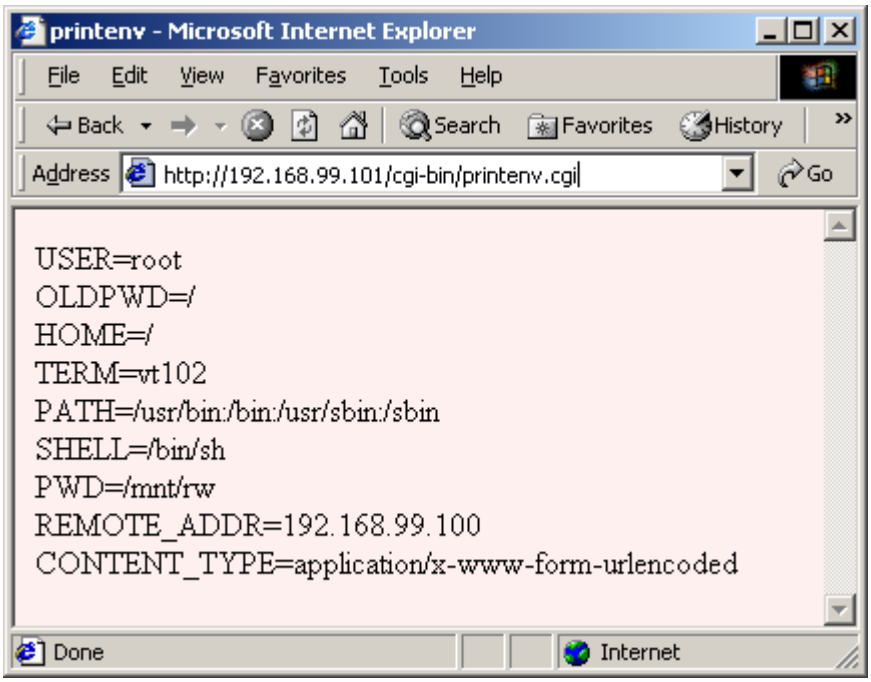
```
#chmod a+x printenv.cgi
```

Programın konsolda çalıştırıyoruz.

#./printenv. cgi

```
<html>
<head><title>printenv</title></head>
<body bgcolor = "#fff0f0">
USER=root<br>
MAIL=/var/spool/mail/root<br>
HOME/root<br>
LOGNAME=root<br>
TERM=vt102<br>
PATH=/bin:/usr/bin:sbin<br>
SHELL=/bin/sh<br>
PWD=/mnt/rw/tmp<br>
</body>
</html>
#
```

Program browser'da çalıştığında aşağıdaki çıktıyı alırız.



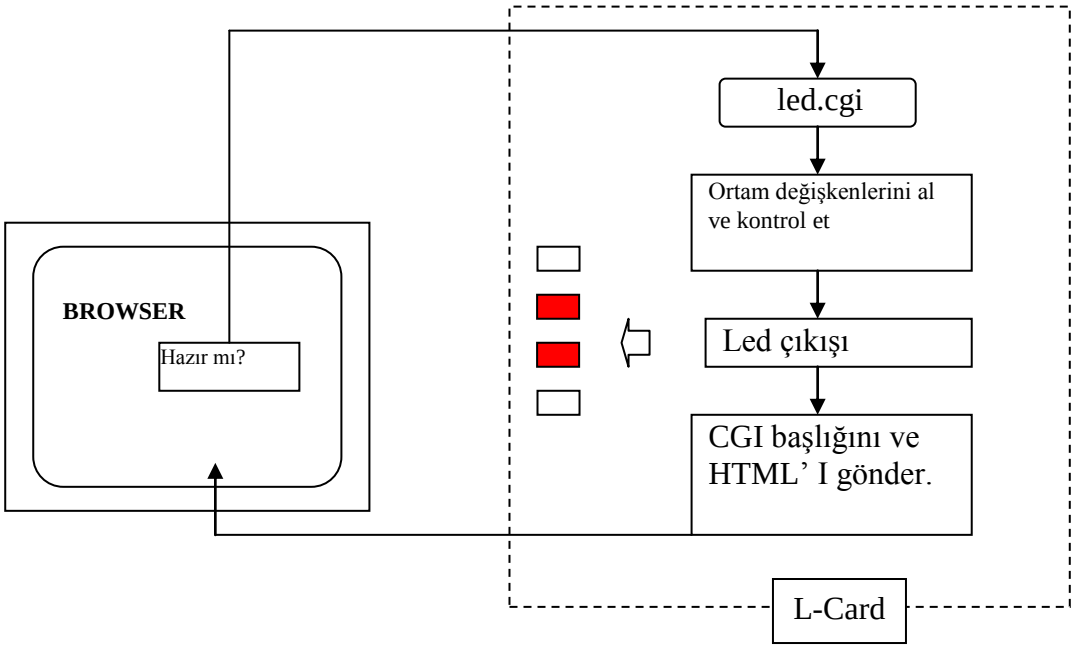
Resim 3.6: Program çıktısı

3.4.1.1. Konsol ve Browser Görüntülerinin Farkı

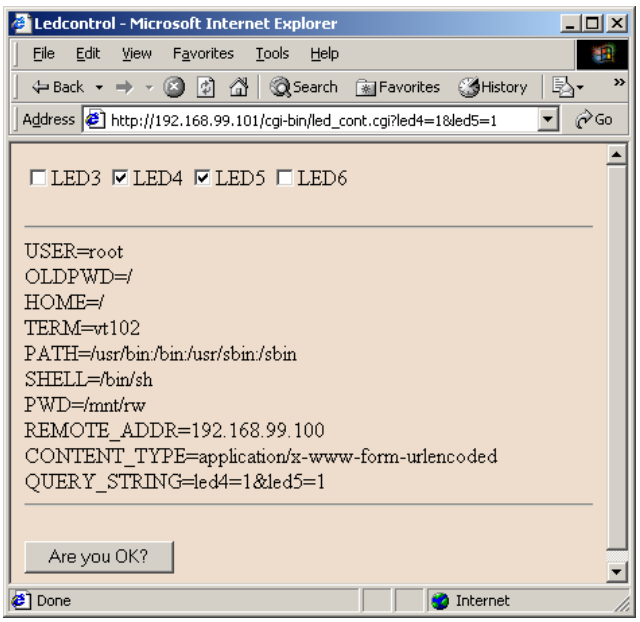
L-Card üzerinden çalıştırıldığında bütün bilgiler görüntülenir. Fakat bir browser tarafından çağırıldığında sadece browser tarafından istenilen gerekli bilgiler gönderilir. Bütün bilgilerin gönderilmemesinin sebebi sunucu güvenliğidir.

3.4.2. Led Kontrol Programı

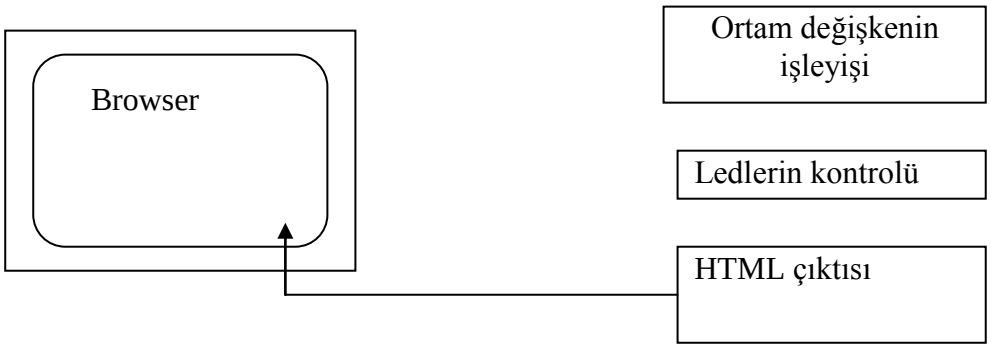
Uzun bir anlatımdan sonra şimdi Led kontrol cgi programını yapalım.



Şekil 3.3: Led Kontrol Programı çalışması



Resim 3.7: Led kontrol programı çıktısı



Program kodları:

```
#include <stdio.h>
#include <fcntl.h>
#include <stdlib.h>
#include <string.h>

#define LED_MIN 3
#define LED_MAX 6

int baLed[LED_MAX+1];

char *query_string_get()
{
        char *pcQs=NULL;
        char *pcCl=getenv("CONTENT_LENGTH");

        if(pcCl==NULL){
        pcQs=getenv("QUERY_STRING");
                        }

if(pcQs==NULL)pcQs= "";
return pcQs;
}
```

Ortam değişkeni PcQs, QUERY_STRING' in pointer' ıdır.

getenv ( ) fonksiyonu Linux' te sadece gerekli ortam değişkenlerini alır.

```
void led_set()
        {
        int i,h;
        char devname[40];

        for(i=LED_MIN;i<=LED_MAX;i++){
        sprintf(devname, "/tmp/led%d",i);
        h=open(devname,O_WRONLY);
        if(h<0){
        printf("Content-type: text/html\n\n");
        printf("open %s failed.<br>\n", devname);
        exit(0);
        }
        else{
        write(h,baLed[i]? "0":"1" ,1);
        close(h);
             }
        }
}
```

led3~led6 için aygıt dosyasının hazırlandığı bölümdür.

Aygıt dosya adı işlemi

Aygıt dosyası yoksa hata olarak göster.

Aygıt dosyası var ise baLed isimli LED dizisine 1 veya 0 gönder.

```
int led_test(char *qs,int n)
{
        char *p,name[10];
        int len;
        sprintf(name, "led%d",n);
        len=strlen(name);
        for(p=qs;p!=NULL; ){
        if(strncmp(name,p,len)==0)
        return 1;
        p = strchr(p, '&');
        if(p!=NULL) p++;
        }
        return 0;
        }
```

QUERY_STRING içinde led3~led6'nın bilgisinin var olup olmadığının kontrol edildiği bölümdür.

Dosya adı oluşturuluyor.

Dosya adı ve QUERY_STRING karşılaştırması yapılıyor.

```
void led_parse()
{

        char *qs = query_string_get();
        int i;
        for(i=LED_MIN;i<=LED_MAX;i++){
        if(led_test(qs,i)) baLed[i]=1;

        else baLed[i]=0;
        }
}
```

Bu kısım diziye 1 veya 0 bilgisi gönderir.

QUERY_STRING içinde led3~led6 var ise diziye 1 bilgisini gönder.

```
static char *header[]={
        "Content-type: text/html",
"",
"<html>",
"<head><title>Ledcontrol</title></head>",
"<body bgcolor=\"#eeddcc\">",
"<form action=\"led_cont.cgi\" method=\"get\">",
NULL,
};
```

HTML başlık tanımlaması

```
static char *trailer[]={
"<p>",
"<input type=\"submit\" value=\"Are you OK?\">"
"</form>",
"</body>",
"</html>",
NULL,
};
```

Görüntülenecek veri tanımlaması

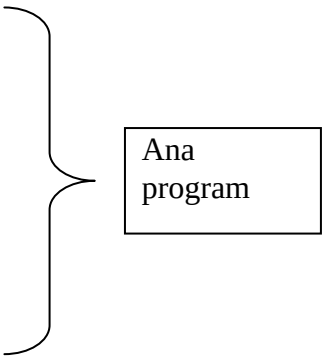
```
void dump_env()
{
extern char **environ;
char **pp;
printf("<p><hr>\n");
for(pp=environ;*pp;pp++)
        printf("%s<br>",*pp);
printf("<hr><p>\n");
}

void html_out()
{
        char **pp;
        int i;

        for(pp=header;*pp;pp++)
        printf("%s\n",*pp);

        for(i=LED_MIN;i<=LED_MAX;i++){
        printf("<input type=\"checkbox\" name=\"led%d\" value=\"1\"",i);
        if(baLed[i])
        printf("checked");
        printf(">LED%d\r",i);
        }

        dump_env();

        for(pp=trailer;*pp;pp++)
        printf("%s\n",*pp);
}
```

Ortam değişkeni tanımlaması

HTML dosyası hazırlama bölümü

Ortam değişkenlerini görüntüle

```
int main()
{
led_parse();
led_set();
html_out();
return 0;
}
```



Açıklama

Programı derliyoruz.

```
Administrator @ JICA-3BAC1EFB9E ~/data
$ Mipsel-Linux-Gcc  -O  -static –o  led_cont.cgi  led_cont.c  -s
```

Programı L-Card'a aktarıyoruz.

```
# cd /tmp
# wget ftp://usr:pass@192.168.99.100 / led_cont.cgi
led_cont.cgi  100% |*****************************|  149 KB    00:00 ETA
```

Çalıştırılabilir izin ekliyoruz.

```
#chmod a+x led_cont.cgi
```

Programı konsolda çalıştırıyoruz.

```
# cd /tmp
gpio.o
#lsmod
Module     Size   Used  by
gpio       2496  0
```

Device file error

```
#insmod gpio.o
#mknod  /tmp/led3  c  17  29
#mknod  /tmp/led4  c  17  31
#mknod  /tmp/led5  c  17  8
#mknod  /tmp/led6  c  17  2
```

UYGULAMA FAALİYETİ

Gömülü sistem kartının internet üzerinden kontrolünü yapınız.

İşlem Basamakları	Öneriler
➢ Aşağıda verilen kodları yazarak L-Card'da çalıştırınız.	➢ Kodları yazarken not defteri ya da "vi" editörünü kullanabilirsiniz.

1. Aşağıdaki HTML uygulamasını yazıp çalıştırınız.

```
<html><head>
<title> Bu Benim Sayfam</title></head>
<body>
</body></html>
```

2. Aşağıdaki HTML uygulamasını yazıp çalıştırınız.

```
<html>
<head><title>Metinleri Kalın Yazdırmak</title>
</head><body>
Endüstriyel<b>Otomasyon Teknolojileri</b>Bölümü
<br>
<b>Denetim Sistemleri</b>Dersi
</body>
</html>
```

3. Aşağıdaki CGI uygulamasını yazınız. Kaydediniz. Derleyiniz. L-Card üzerinde çalıştırınız.

```
#include<stdio.h>
int main( )
{
      printf("Content-type: text / plain\r\n");
      printf("\r\n" );
      printf("İlk CGI Programım\r\n");
      printf("Herkese Başarılar!!!\r\n" );
      return 0;
}
```

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Herhangi bir web sayfasını, web sayfası görüntüleyicimizden açmaya çalıştığımızda bağlandığımız ve bilgisayarımıza web sayfasının dosyalarını gönderen, böylece web sayfalarını görüntüleyebilmemizi sağlayan sunucular hangileridir?

A) Ftp sunucuları
B) DNS sunucuları
C) Dosya sunucuları
D) Web sunucuları

2. Aşağıdakilerden hangisi web sunucusu ve web sayfası görüntüleyicisi arasında kullanılan protokolün ismidir?

A) TCP/IP
B) UDP
C) HTTP
D) FTP

3. Aşağıdakilerden hangisi web sayfası yapmak için kullandığımız bir programlama dilidir?

A) C++
B) HTML
C) Basic
D) Pascal

4. Aşağıdakilerden hangisi kullanıcı ile etkileşimli, aktif web sayfaları tasarlamak için kullandığımız bir programlama dilidir?

A) HTML
B) CGI
C) SQL
D) Clipper

5. HTML dilinde web sayfası hazırlarken yazıların karakterini büyüklüklerini, renklerini, ayrıca web sayfaları ile ilgili artalan rengi gibi birtakım özellikleri değiştirmek için kullanılan etiketlere ne isim verilir?

A) Tag
B) Compiler
C) Command
D) Frame

6. Yazıları kalınlaştırmak, yazı fontunu değiştirmek, başlık boyutunu ayarlamak için etiketlerin içine yazılan harfler aşağıdaki seçeneklerin hangisinde doğru sırada verilmiştir?

A) Font,br,b
B) H*,font,b
C) Br,h*,b
D) B,Font,H*

7. CGI dilinde yazılan programların uzantısı, aşağıdakilerden hangisidir?

A) PHP
B) ASP
C) CGI
D) HTML

8. Veri girişi için kullanılan HTML etiketi aşağıdakilerden hangisidir?

A) <Form>
B) <Head>
C) <HTML>
D) <Body>

9. GET metodunda ortam değişkeni nereden okunur?

A) CONTENT_STRING
B) QUERY_STRING
C) CONTENT_LENGTH
D) QUERY_LENGHT

10. POST metodunda ortam değişkeni nereden okunur?

A) CONTENT_STRING
B) QUERY_STRING
C) CONTENT_LENGTH
D) QUERY_LENGHT

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise "Modül Değerlendirme"ye geçiniz.

MODÜL DEĞERLENDİRME

Bu modül kapsamında aşağıda listelenen davranışlardan kazandığınız becerileri Evet, kazanamadığınız becerileri Hayır kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. L Card yapısını anladınız mı?		
2. L Card bilgisayar bağlantısını yaptınız mı?		
3. LCard'ı açıp kapatabildiniz mi?		
4. Gömülü sistem kartı temel ayarlarını yaptınız mı?		
5. Gömülü sistem kartına program yazdınız mı?		
6. Kernel modunu ve User modunu kullandınız mı?		
7. Gömülü sistem giriş çıkış portlarını kullandınız mı?		
8. Aygıt sürücüsünü kurabildiniz mi?		
9. Web sayfası üzerinden donanım kontrolünü yaptınız mı ?		
10. CGI dilini kullandınız mı ?		

DEĞERLENDİRME

Değerlendirme sonunda "Hayır" şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız "Evet" ise bir sonraki modüle geçmek için öğretmeninizle iletişim kurunuz.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ 1'İN CEVAP ANAHTARI

1	**B**	**6**	**B**
2	**A**	**7**	**A**
3	**D**	**8**	**C**
4	**C**	**9**	**C**
5	**C**	**10**	**B**

ÖĞRENME FAALİYETİ 2'NİN CEVAP ANAHTARI

1	**A**	**6**	**A**
2	**D**	**7**	**B**
3	**C**	**8**	**C**
4	**A**	**9**	**B**
5	**A**	**10**	**B**

ÖĞRENME FAALİYETİ 3'ÜN CEVAP ANAHTARI

1	**D**	**6**	**D**
2	**C**	**7**	**C**
3	**B**	**8**	**A**
4	**B**	**9**	**B**
5	**A**	**10**	**C**

KAYNAKÇA

- NAZMAN Mustafa, Murat AKDOĞAN, **Endüstriyel Ürün Tasarım Ders Kitabı**